# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# SUPPLEMENTAL TO THESIS (4 OF 10: TRAC-MONTEREY 17A SURVEY ANALYSIS (SENTIMENT_TEXT ANALYSIS))

**RECRUITING THE CYBER LEADER: AN EVALUATION OF THE HUMAN RESOURCE MODEL USED FOR RECRUITING THE ARMY'S "CYBER OPERATIONS OFFICER"**

by

Wallace C. Nicholson
Sean A. Gibbs

September 2017

| | |
|---|---|
| Thesis Co-Advisors: | Steve Mullins |
| | Alejandro Hernandez |
| Second Reader: | William Hatch |

**Approved for public release. Distribution is unlimited.**

# 17A Survey Analysis Support

*MAJ Nate Parker, TRAC-MTRY, nparker@nps.edu*

*September 13, 2017*

## Overview

The purpose of this write-up is to document the text analysis support provided by the author to the 17A Survey Thesis team. We include all of the code needed to support future reproduction of the analysis and encourage future research to move the ball further forward. For brevity this write-up only covers the analysis of Question 38, the results for Question 39 can easily be obtained by substituting variable names.

## Setting up the R Environment

Here we load all the required packages. Note that in this instance the *rJava* package requires us to point to a specific *jre* file - the individual system requirements to implement this step will vary. We also set the random seed to support reproduction of several of the network graph we will generate later.

```
library(tibble)
library(tm)
library(wordcloud)
library(dplyr)
library(RColorBrewer)
library(widyr)
library(tidytext)

library(ggplot2)
library(igraph)
library(ggforce)
library(ggraph)



options(java.home="C:\\Program Files\\Java\\jdk1.8.0_91\\jre")
library(rJava)
library(ngram)
library(qdap)

set.seed(1234)
```

## Loading and subsetting data

Here we load the data from a pre-formatted csv. The 17A Survey Thesis team previously conduct manual coding of each response to the questions of interest (Q38 and Q39) to identify the response as dealing with Selection, Recruitment, or None. We then subset the full data set to those response corresponding to each of the coded labels for Question 38.

```
data <- read.csv("C:\\Users\\nparker\\Desktop\\19A\\19A_data.csv",
                 stringsAsFactors = FALSE)

####Subset the data
```

```
sel.38.data <- data %>% filter(Response.Category == "SELECTION",
                               Question == "Q38") %>% select(Response)

recruit.38.data <- data %>% filter(Response.Category == "RECRUITMENT",
                                   Question == "Q38") %>% select(Response)

none.38.data <- data %>% filter(Response.Category == "NONE",
                                Question == "Q38") %>% select(Response)
```

## Pre-processing the text responses.

Here we define a function that will execute our pre-processing of the dataframe object into the Corpus object that the *wordcloud* package expects as input. As part of the pro-processing we first transform all characters to lower case, remove punctuation, remove extra whitespace, and remove the stopwords. In addition to the standard SMART and english stopword sets we also remove the following words: technical, process, and branch. Early exploratory analysis indicated that these words were generically structural within the dataset and their processes only introduced unnecessary noise in the dataset.

The function takes a concatenated vector of the responses of interest, the column names of the vectors, and any additional stopwords. The output is a term-document matrix (tdm).

```
extraStopwords <- c("technical", "process", "branch")

corpFunction <- function(input_data, c_list, extraStopwords){

  c <- VCorpus(VectorSource(input_data))

  c <- tm_map(c, content_transformer(tolower))
  c <- tm_map(c, removePunctuation)
  c <- tm_map(c, content_transformer(stripWhitespace))

  c <- tm_map(c, removeWords, tm::stopwords("SMART"))
  c <- tm_map(c, removeWords, tm::stopwords("english"))
  c <- tm_map(c, removeWords, extraStopwords)
  c <- tm_map(c, PlainTextDocument)

  cloud.tdm = TermDocumentMatrix(c)
  cloud.tdm.m = as.matrix(cloud.tdm)

  colnames(cloud.tdm.m) <- c_list

  return(cloud.tdm.m)

}
```

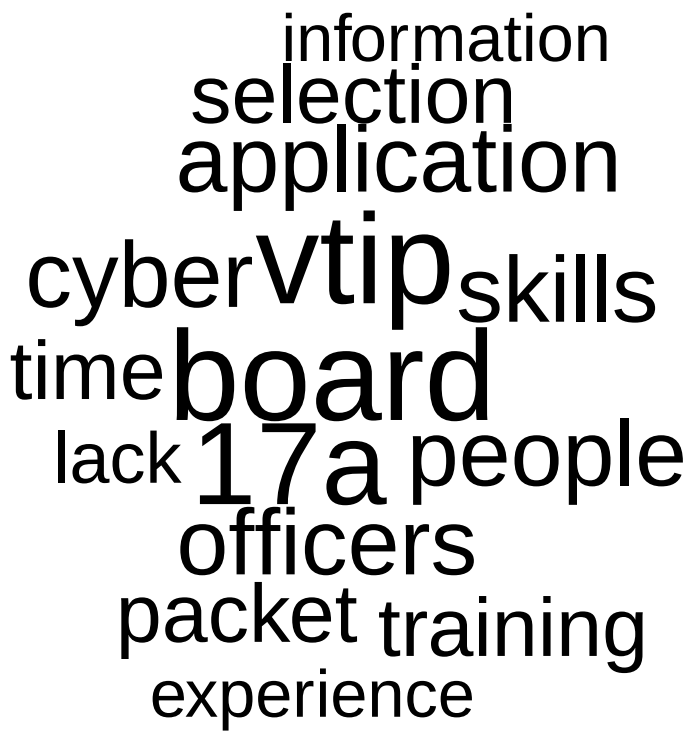
## Generating Comparison and Commonality Wordclouds.

Here we use the *wordcloud* package to generate the comparison and commonality wordclouds.

```
cloud.38 <- c(sel.38.data, recruit.38.data)

c_list.38 <- c("Selection", "Recruiting")
```

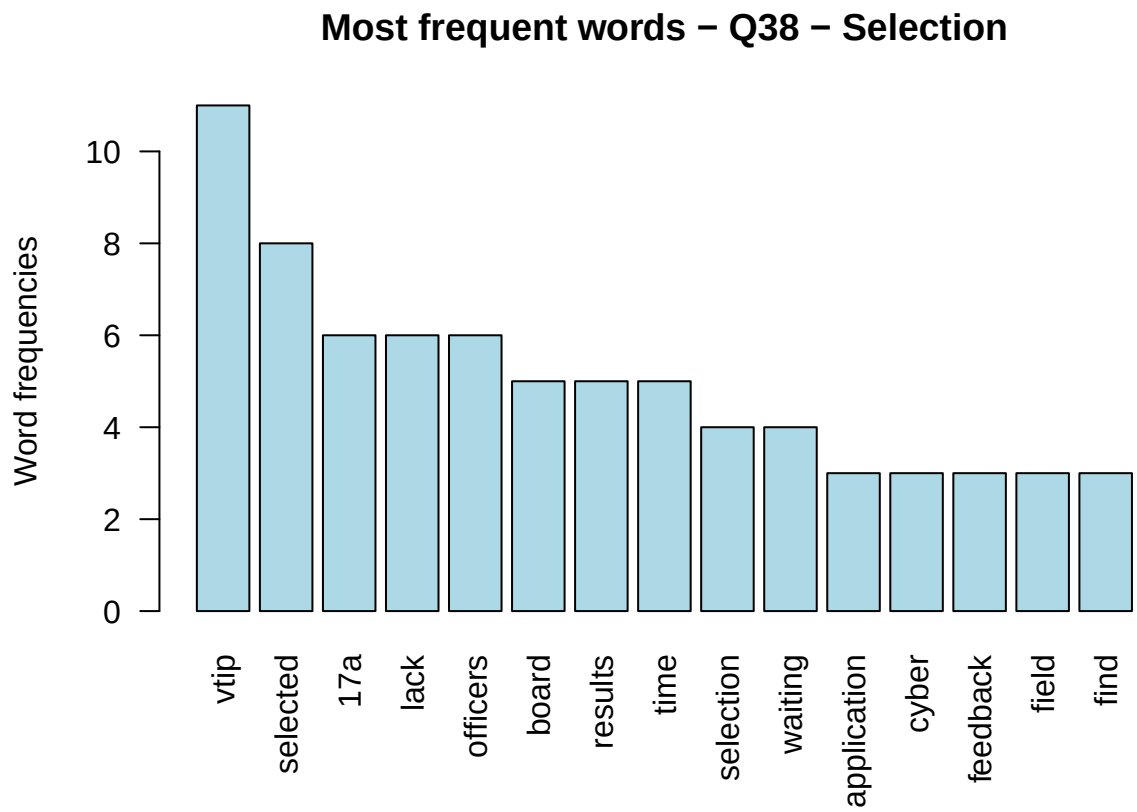
```
tdm.38.m <- corpFunction(cloud.38, c_list.38, extraStopwords)

comp.38 <- comparison.cloud(tdm.38.m, min.words = 4, max.words = 50, random.order= FALSE,
                            title.size = 2, rot.per = 0)
```

```
commmon.38 <- commonality.cloud(tdm.38.m, min.words = 4, max.words = 15,
                                random.order= FALSE)
```

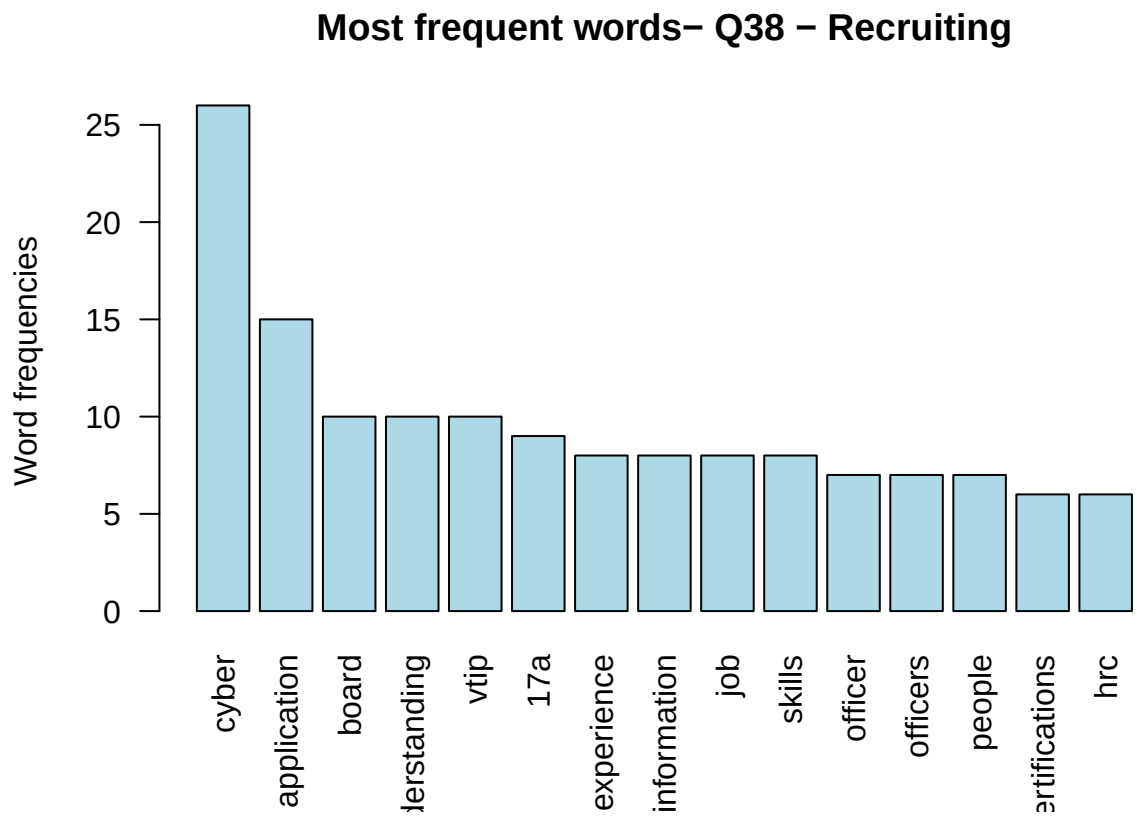



These visualization give a qualitative sense of the data but lack some of the quantitative data that some may find helpful. To support this requirement we also generate bar graphs of the most common words.

```
sel.38.d <- data.frame(word = rownames(tdm.38.m), freq=tdm.38.m[,1])

sel.38.d <- sel.38.d[order(-sel.38.d$freq),]

rec.38.d <- data.frame(word = rownames(tdm.38.m), freq=tdm.38.m[,2])

rec.38.d <- rec.38.d[order(-rec.38.d$freq),]

sel.38.bar <- barplot(sel.38.d[1:15,]$freq, las = 2, names.arg = sel.38.d[1:15,]$word,
                      col ="lightblue", main ="Most frequent words - Q38 - Selection",
                      ylab = "Word frequencies")
```




```
rec.38.bar <- barplot(rec.38.d[1:15,]$freq, las = 2, names.arg = rec.38.d[1:15,]$word,
                      col ="lightblue", main ="Most frequent words- Q38 - Recruiting",
                      ylab = "Word frequencies")
```




### Correlation and co-occurance

We now leverage so work conducted by LTC Melanie Vinton to visualize the co-occurrence and correlation of words within responses. His full documentation is published at http://dscoe.army.mil/files/9415/0170/2779/text_correlation.html.

We first develop two functions (one for co-occurrence and one for correlation) to process that data into the format our visualization calls expect. Here we use the *tidytext* package to replicate the same pre-processing steps that we used the *tm* package to complete in the wordcloud functions above.

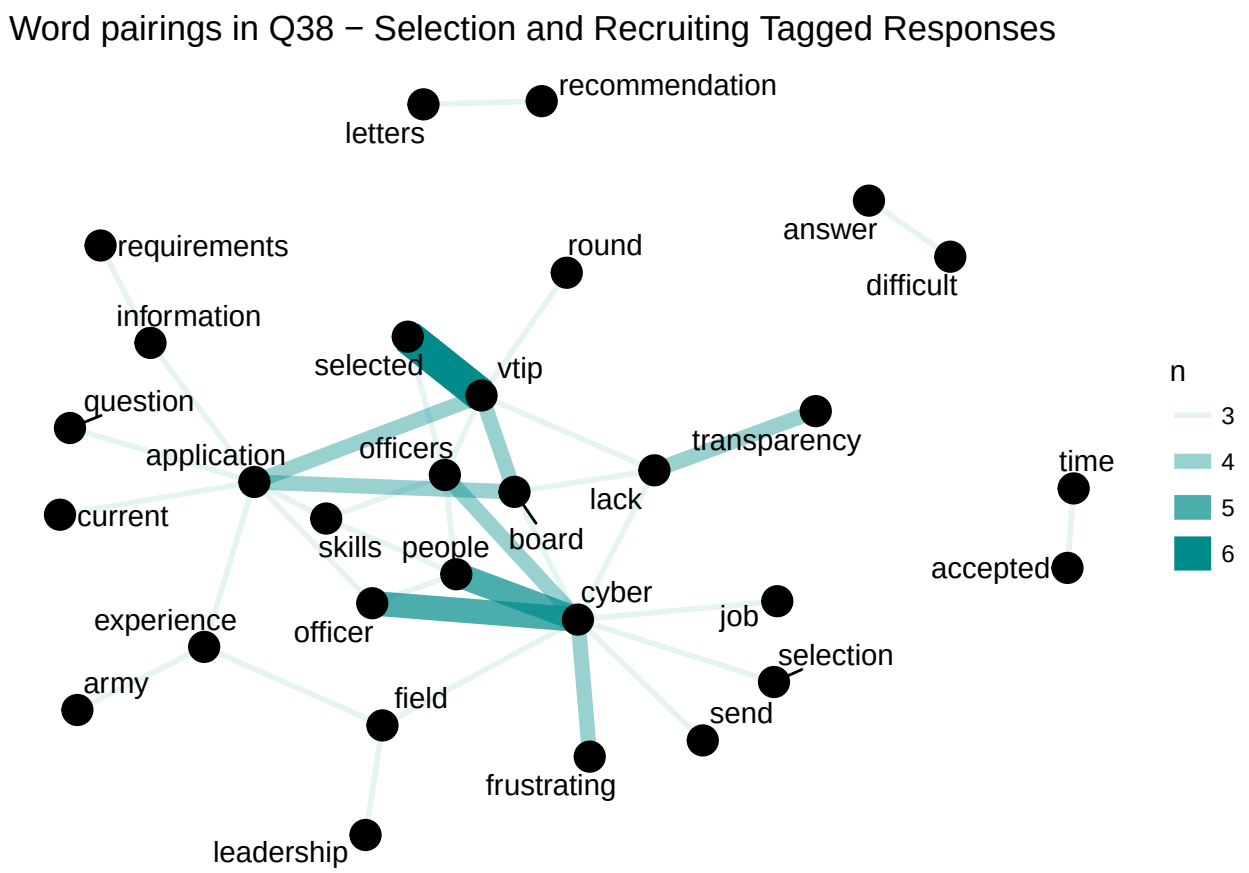
Both functions take the following as inputs: a single vector of freetext responses, the base set of stopwords, and any extra stopwords. The correlation function also includes a cutoff value as an input. Note that now we are using all responses to Question 38 that had either a Selection or Recruiting tag.

```
cooccuranceFunction <- function(input_data, stops, extraStopwords){

  d <- input_data

  d$id <- seq(1,nrow(d))

  d.tidy <- unnest_tokens(tbl = d, output = word, input = Response)

  d.tidy <- filter(d.tidy, word != "")

  d.tidy <- dplyr::anti_join(d.tidy, stops)
```

```r
  d.tidy <- dplyr::anti_join(d.tidy, data.frame(word = extraStopwords))

  d.tidy.pairs <- pairwise_count(d.tidy, word, id, sort = TRUE, upper = FALSE)

  return(d.tidy.pairs)

}

corFunction <- function(input_data, stops, extraStopwords, cutoff){

  d <- input_data

  d$id <- seq(1,nrow(d))

  d.tidy <- unnest_tokens(tbl = d, output = word, input = Response)

  d.tidy <- filter(d.tidy, word != "")

  d.tidy <- dplyr::anti_join(d.tidy, stops)

  d.tidy <- dplyr::anti_join(d.tidy, data.frame(word = extraStopwords))

  text_cor <- filter(group_by(d.tidy, word), n() > cutoff)

  text_cor2 <- pairwise_cor(text_cor, word, id, sort = TRUE, upper = FALSE)

  return(text_cor2)

}

### Co-occurance Q38
Q38 <- rbind(sel.38.data, recruit.38.data)

Q38.pairs <- cooccuranceFunction(Q38, stop_words, extraStopwords)
```

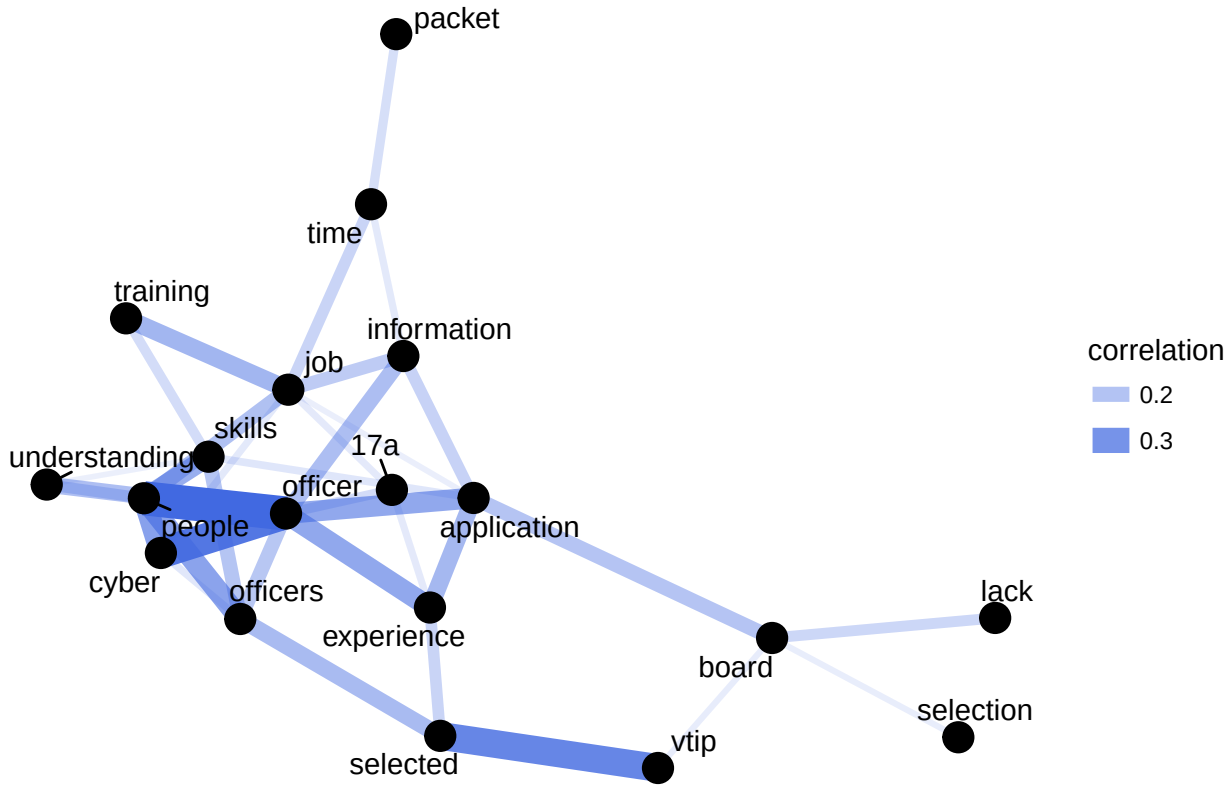
```
## Joining, by = "word"
## Joining, by = "word"
```

```r
Q38.pairs.2 <- filter(Q38.pairs, n > 2)

Q38.g <- graph_from_data_frame(Q38.pairs.2)

Q38.g2 <- ggraph(Q38.g, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n, edge_width = n), edge_colour = "cyan4") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  ggtitle("Word pairings in Q38 - Selection and Recruiting Tagged Responses")+
  theme_void()

Q38.g2
```




```
###Q38 correlation

Q38.cor <- corFunction(Q38, stop_words, extraStopwords, 7)
```

```
## Joining, by = "word"
## Joining, by = "word"
```

```
Q38.cor2 <- filter(Q38.cor, correlation > 0.1)

Q38.c <- graph_from_data_frame(Q38.cor2)

Q38.c2 <- ggraph(Q38.c, layout = "fr") +
  geom_edge_link(aes(edge_width = correlation, edge_alpha = correlation),
                 edge_colour = "royalblue") + geom_node_point(size = 5) +
                 geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  ggtitle("Word Correlations for Q38 - Recruiting and Selection Tags") +
  theme_void()

Q38.c2
```

Word Correlations for Q38 − Recruiting and Selection Tags